Engineer

コード品質

コードが正しく動作し、理解や変更がしやすい度合い

どのようなものか

コード品質とは、ソースコードが意図したとおりに動き、担当者が理解、検証、変更しやすい度合いのことです。請求額を計算する小さな関数で考えてみましょう。ある実装は、開発時に想定した注文の流れをたどると正しい額を返します。別の実装は、通貨を明示し、丸めの規則を一か所に置き、返金の場合を区別します。どちらも標準的な注文のテストには合格するかもしれません。請求を受けた人から金額の問い合わせが来たとき、調査の出発点が分かりやすいのは後者です。

正しさは、不正な入力、再試行、処理が途中で止まる場合にも問われます。読みやすさは、次に読む人が重要な設計判断をたどれるかどうかに表れます。変更のしやすさには、新しい要求がどこまで波及するか、結果を確かめられるかが関わります。短い関数でも税額を誤ることがあります。複雑な関数は現在の条件では正しくても、次に通貨を追加するとき、検証が難しいかもしれません。

コードレビューは、実際の変更に即してこうした性質を評価する方法です。品質ゲートは、決められた検査に合格しない変更が次の工程へ進むのを止めます。技術的負債は、設計上の判断が将来の変更にもたらす負担に注目する考え方です。コード品質が指すのはコード自体の状態です。Google のレビュー基準では、個々の変更によってコードベース全体の健全性が向上するかを問います。その視点を採れば、日々の修正で何を変えるべきかを話し合えます。

概念を判断に生かす

「コード品質とは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 3社の間で受け渡す代わりに、1つのチームがアイデアのスケッチから、お客様がログインして使える製品まで担当します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 GIS地図アプリの開発を主導し売上を10倍に — 10x 地図製品の開発を主導して実現した売上成長

GIS地図アプリの開発を主導し売上を10倍に お問い合わせ →

なぜ重要なのか

品質の差がはっきりするのは、要求が変わったときです。見積もりを作成し、請求書を発行し、その後に返金するサービスを考えます。各処理が同じ割引条件を別々に実装していれば、新しいキャンペーンのたびにすべての実装を探して更新する必要があります。変更漏れは、初めて返金が必要になるまで表面化しないかもしれません。一つの規則を既知の場所に置き、各処理が入力を明示すれば、変更の影響を見積もりやすくなり、追加すべきテストも判断できます。

必要な証拠はリスクによって異なります。金額の計算では、行ごとに丸める場合と合計してから丸める場合で答えが異なる例を選びます。再試行できる処理では、二度目の要求で再び請求されるかを確かめます。決済のたびに実行される計算なら、キャッシュを追加する前に、想定負荷でどれだけ時間がかかるかを計測します。どの観察も、条件と結論を結び付けて初めて判断材料になります。

引き継ぎにも品質が表れます。人は担当を変わり、レビューを引き受けた人がモジュールの経緯を知っているとは限りません。意味が伝わる名前、実行できる具体例、珍しい制約についての短い説明があれば、元の担当者を待たずに重要な判断を確認できます。これは新しいメンバーの参加時にも、夜間に障害を調べるときにも役立ちます。現在の担当者から見て明快かどうかだけでは、引き継ぎのしやすさは分かりません。事情を知らない人が前提をたどって結論を確認できることが大切です。

実際の活用例

仮に、税率ごとに税額を集計してから丸める規則があるとします。請求書の二つの明細について、丸める前の税額がそれぞれ 0.005 通貨単位だとしましょう。ちょうど中間の値を切り上げて明細ごとに小数点以下第 2 位に丸めると、合計は 0.02 になります。先に合計して一度だけ丸めれば、0.01 です。レビューの場で電卓がなくても確認できるよう、わざと小さな数字にしています。実際の請求書でどちらの処理を採るかは、製品の要求と適用される規則によって決まります。

まず、関数が受け取る値の単位と計算の順序を明記します。途中の値は必要な精度を保てる形式で扱います。税率ごとの集計をコードから読み取れるようにし、合意した箇所で丸めます。前述の値をテストに使えば、二つの規則を見分けられ

ます。二種類の税率が混ざる例も用意すると、将来の書き換えで税率別の集計が一つの総額に置き換わる不具合を検出できます。返金に別の規則を使うなら、それを示す例も必要です。

修正を確認するときは、利用者に表示する額、書き出す額、返金時に渡す値を追います。過去の請求書では、発行時に保存した額を保ちます。計算だけを取り出したテストで、既知の入力に対する答えを確認できます。一連の処理を通すテストでは、実際の呼び出し元が正しい単位の値を渡し、意図した位置で計算が行われることを調べます。

最後のチケットを一度だけ販売するためのノートは、在庫について同じ考え方を示しています。常に成り立つべき数量の関係を記し、処理が同時に走る場合のテストを挙げ、その関係を稼働中も監視する案を示しています。何を維持する必要があるのか式で表すと、複数の処理経路にまたがる変更も点検できます。

問題が起きるところ

測れる指標を保証と取り違えることがあります。カバレッジからは、テストがどの行や分岐を通ったかが分かります。誤った丸め規則を期待値に書いても、テストは合格し得ます。複雑さの指標は、詳しく読むべき場所を探す手掛かりになります。しかし、一目で分かる比較をいくつもの曖昧な補助関数に分ければ、障害調査は難しくなります。新しい指標を採る前に、どんな欠陥を見つけたいのかを決め、しきい値の両側の例を読んでみます。

整理に熱心なあまり、優先順位を見失うこともあります。めったに使わない報告書の変数名と、すべての決済に使う条件式では、曖昧さの影響が違います。実際に見つかった不具合やリスクを記録し、必須の修正と後からできる改善を分けます。書式に関する意見の違いなら、共有ルールを決めて自動整形に任せられることがあります。

部分的な改善で、別の場所に問題が移る場合もあります。複数の処理が同じ商慣行に従うときは、割引を共通の関数にまとめる意味があります。一方、契約更新、販売促進、返金で異なる規則を用いるなら、一つの関数に推測させると設計が見えにくくなります。共通する入力と条件を調べ、安定した部分だけを共有し、違いをコードに表します。次に読む人が規則を説明できるか、テストで結果を確認できるかを見直せば、修正の妥当性を判断できます。

参考資料

- Google のコードレビューの承認基準
- Google のレビューで確認する項目
- 『Software Engineering at Google』の単体テストの章

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「コード品質とは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/code-quality/